

Annotasi Adaptif Patch Citra Apusan Dahak Bakteri Tahan Asam Berbasis Nilai Eigen Laplacian Graf

Author Name and NIM^{1,2}

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

¹13524114@std.stei.itb.ac.id, ²mirzatsabita72@gmail.com

Abstract—Anotasi akurat pada citra apusan dahak bakteri tahan asam (BTA) masih menjadi tantangan karena adanya noise, objek yang terfragmentasi, dan variabilitas kualitas pewarnaan pada tingkat patch. Studi ini mengusulkan kerangka kerja anotasi berbasis graf adaptif eigen yang menyempurnakan hasil segmentasi biner menggunakan sifat geometri Laplacian. Anotasi biner pertama kali dihasilkan menggunakan operator logika (RGB, OR, AND, XOR, dan XNOR), kemudian ditransformasikan menjadi graf tingkat piksel dengan beberapa skema konektivitas. Langkah pasca-pemrosesan adaptif diperkenalkan, di mana nilai eigen Laplacian terkecil kedua (λ_2) dan jumlah nilai eigen nol digunakan untuk menilai konektivitas graf dan menentukan penyempurnaan iteratif serta strategi penghubungan. Evaluasi eksperimental pada citra patch BTA menunjukkan bahwa mekanisme adaptif eigen yang diusulkan secara konsisten meningkatkan kualitas segmentasi di semua operator logika. Baik penilaian visual kualitatif maupun analisis kuantitatif menggunakan metrik Intersection over Union (IoU) mengkonfirmasi bahwa metode M-AND mencapai perilaku yang paling stabil, konvergen dalam satu iterasi di semua sampel dengan menghasilkan anotasi yang secara visual paling dekat dengan ground truth-nya. Secara kuantitatif, M-AND mencapai IoU rata-rata tertinggi (0,74), yang sesuai dengan peningkatan relatif 23.33 % dibandingkan dengan metode dasarnya. Hasil ini menunjukkan bahwa mengintegrasikan operator logika sederhana dengan kriteria geometri graf menghasilkan strategi anotasi yang efisien, mudah diinterpretasikan, dan kuat. Pendekatan yang diusulkan menawarkan alternatif ringan untuk pemrosesan adaptif dalam analisis citra mikroskopis dan sangat cocok untuk alur kerja anotasi unsupervised dalam aplikasi skrining tuberkulosis.

Keywords— Adaptive, Bakteri Tahan Asam, Eigenvalue, Graph, Laplacian

I. INTRODUCTION

Tuberculosis (TB) di Indonesia masih menjadi penyakit dengan prevalensi terbesar ke-2 di Dunia, dengan jumlah 1.260 jiwa. Screening TB di Indonesia mayoritas menggunakan metode Bakteri Tahan Asam (BTA). BTA menggunakan apusan dahak untuk menentukan tingkat keparahan TBC. Metode apusan dahak yang digunakan di Indonesia masih menganut metode Ziehl-Neelsen (ZN). Dimulai dengan pengambilan dahak pasien, lalu dahak di apuskan ke slide dan dilakukan proses pewarnaan biru.

Bakteri TBC akan mempertahankan warnanya yaitu merah, sedangkan dahak atau selain bakteri akan berubah warna menjadi biru muda. Tingkat keparahan TB didasarkan pada table UATLD, dimana jika dalam 300 layang pandang tidak terdapat BTA, maka deteksinya negative. Jika dalam 300 layang pandang terdapat 1-29 BTA, maka deteksinya scanty. Jika dalam 300 layang pandang terdapat lebih dari 30 BTA, maka deteksinya positif satu. Jika dalam 1 layang pandang terdapat 1-10 BTA, maka deteksinya positif dua. Jika dalam 1 layang pandang terdapat lebih dari 10 BTA, maka deteksinya positif tiga. Perhitungan BTA menjadi krusial, karena jika terjadi kesalahan diagnosis, maka pengobatan akan semakin keras, mahal dan lama, bahkan dapat menyebabkan kematian. Untuk meningkatkan akurasi perhitungan BTA, beberapa peneliti menggunakan CAS (Computer Aided System) yang menggunakan software pengolahan citra untuk menghitung jumlah BTA.

Beberapa peneliti menggunakan system patch (memotong citra besar menjadi citra kecil yang merujuk ke objek), untuk mempercepat proses perhitungan dan meningkatkan hasil akurasi. Perhitungan dimulai dengan melakukan proses segmentasi, dimana proses segmentasi dilakukan untuk memisahkan objek BTA dari latar belakangnya. Proses pembelajaran terbimbing (supervise) menjadi pilihan banyak peneliti, apalagi sudah berkembang teknologi deeplearning. Namun kelemahannya adalah banyak citra medis yang tanpa anotasi, sehingga dibutuhkan metode pembuatan anotasi tapi dari proses *un-supervise*. **Proses pembuatan anotasi mempunyai proses yang sama dengan proses segmentasi.**

Beberapa metode segmentasi *un-supervise* dilakukan oleh peneliti terdahulu, diantaranya adalah region merging cepat [1], min-cut dan max flow[2], random walk[3], region prop [4], random distance [5], graf [6-13] dan deep learning [14]. Metode segmentasi graf menjadi pilihan utama penelitian saat ini, karena graf sangat ringan, tidak membutuhkan operasi yang rumit (seperti operasi citra pada umumnya) dan sangat cocok untuk perhitungan.

Konsep graf yang sudah diteliti oleh peneliti terdahulu diantaranya adalah segmentasi graf berbasis cluster [6], berbasis threshold [7], partisi [8-9], graf non-linear [11], modality [12], dan weakly supervised [13]. Pada segmentasi graf, terdapat 2 masalah utama, yaitu jika ingin menghasilkan graf yang mirip dengan annotasinya, maka waktunya lama, namun jika ingin cepat, maka graf masih jauh dari annotasinya. Hal inilah yang mendasari penulisan artikel ini, Kontribusi yang diberikan pada artikel ini adalah :

1. Memberikan usulan metode adaptive baru menggunakan nilai eigen Laplacian graf, khususnya jumlah eigen nol dan algebraic connectivity (λ_2), sebagai kriteria kuantitatif untuk melakukan segmentasi secara adaptif.
2. Menginfestigasi penggunaannya untuk beberapa operator logika (RGB, OR, AND, XOR, dan XAND).

II. TINJAUAN PUSTAKA

Pembuatan anotasi BTA yang pertama kali dilakukan adalah menggunakan metode segmentasi region merge [1] pada tahun 2004, dimana pada penelitian ini hasil segmentasinya masih kurang bagus (bakterinya terputus-putus), sangat terpengaruh dengan threshold dan intensitas cahaya dari citra BTA.

Tabel 1. Ringkasan Dan Perbandingan Antar Metode Graf

Metode	Spektral	Kekurangan
Jarak random-walk (commute time) untuk clustering [5]	Ya (terkait Laplacian pseudoinverse)	Lebih ke distance transform; bukan pipeline mask→graf→postprocess
Clustering/segmentation berbasis graf [6]	Tidak eksplisit eigen	Lebih klasik; tidak spesifik ke pipeline citra biner patch
Diffusion/thresholding di graf untuk klasifikasi/processing [7]	Ya (graph Laplacian)	Metode matematis kuat tapi setup & interpretasi bisa lebih kompleks
Segmentasi sebagai graph partitioning (Ncut) [8]	Ya (eigen problem)	Umumnya butuh graf berbobot + setting parameter; lebih fokus "partitioning" global
Segmentasi = pemotongan graf jadi komunitas [9]	Tidak wajib eigen	Lebih fokus partisi komunitas daripada evaluasi kualitas segmentasi
Konstruksi graf untuk segmentasi cepat [10]	Ya (terkait spectral clustering)	Fokus speed segmentasi umum; tidak spesifik mikroskop/patch biner
Regularisasi nonlinier p-Laplacian [11]	Ya (spektral/graph PDE)	Lebih kompleks (nonlinear), berat untuk baseline tugas/paper awal
Segmentasi sebagai komunitas graf (modularity) [12]	Bisa terkait spektral	Biasanya graf berbobot/region-level; fokus partisi, bukan eigen-based stopping
Weakly supervised segmentation via communities [13]	Graph/community	Butuh framework learning; lebih kompleks untuk baseline autolabel sederhana
Graph Fourier/Laplacian untuk deteksi anomaly [14]	Ya	Fokus anomaly detection, bukan segmentasi patch/labeling

Walaupun hal itu kemudian diperbaiki menggunakan metode minimal out dan maksimal flow [2], tetap saja hasilnya masih jauh dari yang diinginkan, karena metode ini membutuhkan fungsi energi yang sesuai, dimana implementasinya sangat berat bahkan hanya untuk citra patch kecil. Berikutnya random walk [3] juga diusulkan untuk meningkatkan hasil segmentasi, namun random walk masih membutuhkan marker khusus karena metodenya yang probabilistic, metode ini kurang cocok karena menggunakan metode supervise (membuat anotasi tapi dalam prosesnya membutuhkan anotasi).

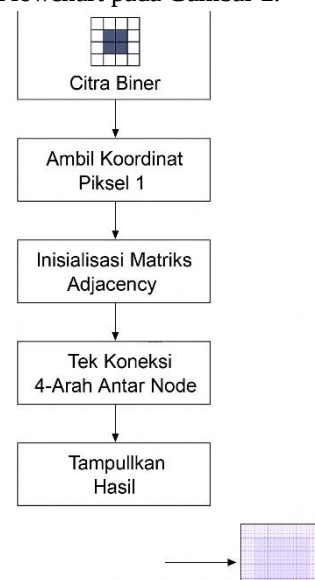
Annotasi un-supervise sekarang lebih mengarah ke graf, karena mulai banyak yang menggunakan Graf Neural Network untuk meningkatkan akurasi dari proses segmentasi. Ringkasan dan perbandingan antar metode graf diperlihatkan pada Tabel 1. Dari table 1, dapat dilihat bahwa metode graf terbaru adalah metode partisi [9], namun metode ini Lebih fokus partisi komunitas daripada evaluasi kualitas segmentasi. Hasilnya masih kurang akurat, walaupun waktu yang dibutuhkan pendek.

Metode dengan iterasi dengan kriteria, memungkinkan untuk peningkatkan performansi hasil segmentasi. Salah satunya adalah dengan menggunakan kriteria nilai eigen laplacian dengan melakukan investasi pada operasi logika.

III. METODE

A. Nilai Eigen Laplacian

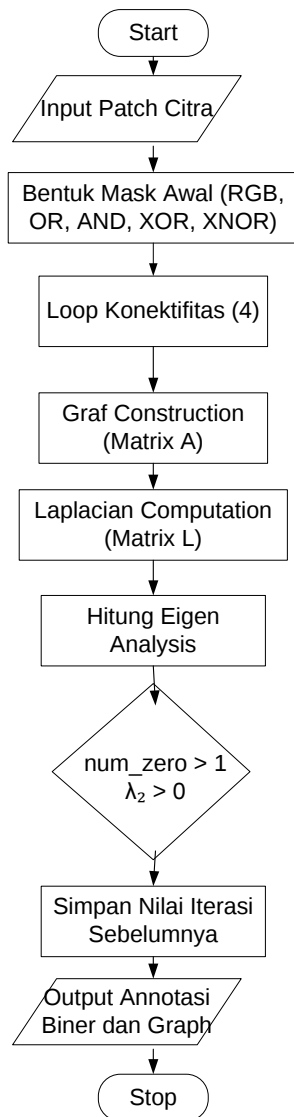
Untuk mendapatkan nilai eigen Laplacian yang pertama kali dilakukan adalah membentuk matrix adjacency (Gambar 1). Untuk membentuk matrix adjacency dilakukan proses design software yang ditunjukkan dengan menggunakan Flowchart pada Gambar 2.



Gambar 1. Flowchart Matrix Adjacency

Dari flowchart yang ditampilkan pada Gambar 2, dimulai dengan pengambilan citra biner dari dataset. Dimana citra biner bisa didapatkan dari citra label, ataupun

citra input yang kemudian diubah menjadi citra biner, ataupun beberapa metode yang lain untuk mengubah citra warna menjadi citra biner.



Gambar 2. Flowchart Eigen Laplacian

Berikutnya dilakukan proses pengambilan nilai koordinat pixel dan dilanjutkan dengan inisialisasi dari matrix adjacency. Hasil tersebut digunakan sebagai input dari pembuatan jaringan koneksi 4 arah dan hasilnya kemudian ditampilkan dalam bentuk graf. Hasil dari matrix adjacency adalah sebuah matrix A yang dapat dilihat pada persamaan 1.

$$A = \begin{bmatrix} a_1 & \dots & a_n \\ \vdots & \ddots & \vdots \\ a_m & \dots & a_{mn} \end{bmatrix} \quad (1)$$

Matrix A pada persamaan 1, merupakan kumpulan nilai 0 dan 1. Berikutnya dilakukan perhitungan derajat dari tiap node menggunakan persamaan 1. Dimana derajat (D) merupakan jumlah nilai 1 dari tiap baris pada matrix A, dimana penempatan kolomnya sesuai dengan posisi barisnya. Matrix D diperlihatkan pada persamaan 2.

$$D = \text{diag} (A) \quad (2)$$

Berikutnya dilakukan proses perhitungan matrix Laplacian (L) yaitu dengan mengambil selisih antara matrix adjacency dengan matrix derajat, yang diperlihatkan pada persamaan 3.

$$L = A - D \quad (3)$$

Kemudian dari matrix L, dicari jumlah nilai nol yang kemudian disebut dengan λ , dan dicari juga nilai eigen atau eigenvalue dari matrix L yang kemudian disebut dengan λ_2 , untuk kemudian disebut dengan eigen analysis. Kriteria iterasi ditentukan oleh eigen analisis-nya. Kriteria iterasi berhenti adalah diperlihatkan pada persamaan 4.

$$\begin{aligned} \text{num_zero} &= 1 \\ \lambda_2 &> 0 \end{aligned} \quad (4)$$

Algoritma Eigen Laplacian sebagai sarana adaptive untuk mendapatkan annotasi graf diperlihatkan pada Gambar 1.

Pada Gambar 1, proses dimulai dari menyiapkan citra input biner yang berbentuk patch (potongan hanya pada lokasi bakterinya saja). Citra biner kemudian dibentuk menjadi annotasi awal (RGB, OR, AND, XOR dan XAND).

B. Operasi Logika OR, AND, XOR dan XAND

Operasi logika yang digunakan diawali dengan mengubah tiap kanal citra yaitu kanal Red (R), kanal Green (G) dan Kanal Blue (B), masing-masing menjadi kanal biner (0 dan 1). Berikutnya melakukan perhitungan untuk:

1. Operasi Citra Warna ke Biner

Proses perhitungan dari citra warna (RGB) menjadi citra biner dilakukan dengan mengubah citra warna ke citra keabuan (gray) menggunakan persamaan 5.

$$I_{gray} = 0.299R + 0.587G + 0.114B \quad (5)$$

Setelah mendapatkan citra keabuan, baru kemudian diubah ke citra biner menggunakan persamaan 6.

$$I_B = \begin{cases} 1, & \text{jika } I_{gray} > \text{threshold} \\ 0, & \text{untuk lainnya} \end{cases} \quad (6)$$

2. Logika OR

Proses logika OR dilakukan dengan menggunakan persamaan 7.

$$M_{OR} = M_R \vee M_G \vee M_B \quad (7)$$

3. Logika AND

Proses logika AND dilakukan dengan menggunakan persamaan 8.

$$M_{AND} = M_R \wedge M_G \wedge M_B \quad (8)$$

4. Logika XOR

Proses logika XOR dilakukan dengan menggunakan persamaan 9.

$$M_{XOR} = M_R \oplus M_G \oplus M_B \quad (9)$$

5. Logika XAND

Proses logika XAND dilakukan dengan menggunakan persamaan 10.

$$M_{XAND} = (M_R \vee M_G \vee M_B) \wedge \neg (M_R \wedge M_G \wedge M_B) \quad (10)$$

Citra hasil operasi logika atau yang disebut dengan annotasi awal, kemudian dimasukkan ke dalam loop

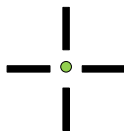
konektivitas, dengan menggunakan 4 konektivitas. Matrix adjacency merupakan matriks yang merepresentasikan sebuah graf, yang terdiri dari himpunan simpul (node) dan himpunan sisi (edge). Persamaan dari Adjacency (A) diperlihatkan pada persamaan 11.

$$A = (V, E) \quad (11)$$

Dimana V merupakan himpunan nodes dan E merupakan himpunan edge, dengan $A \in \mathbb{R}^{n \times n}$, $b=|V|$. Untuk graf yang berasal dari Adjacency Matiks sendiri dilambangkan dengan $A_{a,b}$, dengan rumus yang diperlihatkan pada persamaan 2.

$$A_{(a,b)} = \begin{cases} 1, & \text{jika node a terhubung ke node b} \\ 0, & \text{jika tidak terhubung} \end{cases} \quad (2)$$

Matriks Adjacency juga mempertimbangkan koneksi antara node yang satu dengan node yang lain melalui edge, bisa menggunakan 2 koneksi, 4 koneksi dan 8 koneksi. Untuk 2 koneksi bisa kearah kanan dan kiri atau ke arah atas dan bawah. Sedangkan untuk 4 koneksi (kanan, kiri, atas dan bawah), sedangkan untuk 8 koneksi bisa menggunakan diagonalnya. Representasi dari koneksi antara nodes dan edge pada 4 koneksi diperlihatkan pada Gambar 3.



Gambar 3. Representasi Koneksi Antara Node (lingkaran hijau) dan Edge (Garis hitam) 4 koneksi

Pada penelitian ini, dipilih 4 koneksi, karena untuk menyederhanakan proses, namun representasi graf yang dihasilkan masih mirip dengan citra aslinya. Sebuah contoh kasus, missal sebuah citra biner direpresntasikan pada sebuah matriks input dengan ukuran matriks 5x5 (Gambar 4).

0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	0	0	0	0
0	0	0	0	0

Gambar 4. Matrix Input

Matriks input terdiri dari baris dan kolom, artinya 0 adalah pixel dengan warna hitam dan 1 atau 255 adalah pixel dengan warna putih. Artinya ada 4 node yang kemudian menjadi himpunan nodes (V) (Tabel 2).

Tabel 2. Himpunan Nodes (V)

(Baris/Kolom)	Nama Node
(2,2)	0
(2,3)	1
(3,2)	2
(3,3)	3

Setelah itu dilakukan pembuatan himpunan edge (E), yaitu dengan menentukan simpul bertetangga antara node yang satu dengan node yang lain (Tabel 3). Node tetangga dilakukan dengan 4 koneksi.

Tabel 3. Himpunan Edge (E)

Nama Node	Edge (Node Tetangga)
0	(1,2) artinya adalah (0-1 dan 0-2)
1	(0,3) artinya adalah (1-0 dan 1-3)
2	(0,3) artinya adalah (2-0 dan 2-3)
3	(1,2) artinya adalah (3-1 dan 3-2)

Ketika matrix node (V) dan matrix edge (E) dijadikan 1 tabel, maka hasilnya adalah sebuah Adjacency matriks (A), yang diperlihatkan pada Gambar 5.

Baris/kolom	0	1	2	3
0	0	1	1	0
1	1	0	0	1
2	1	0	0	1
3	0	1	1	0

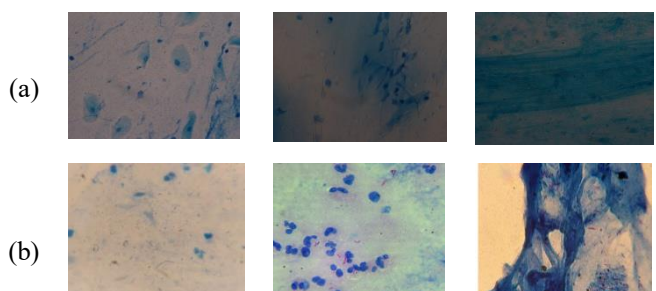
Gambar 5. Matrix Output

Matrix yang terbentuk dari hasil 4 koneksi, adalah berupa matrix adjacency graf (A). Beikutnya dilakukan proses komputasi laplace, yang diawali dengan pembentukan matrix derajat (D) sesuai persamaan 2, yang kemudian digunakan untuk mencari matrix Laplace (L), yang ditunjukan pada persamaan 3. Berikutnya dilakukan proses analisis eigen, yang dilakukan dengan persamaan 4.

C. Eksperimen

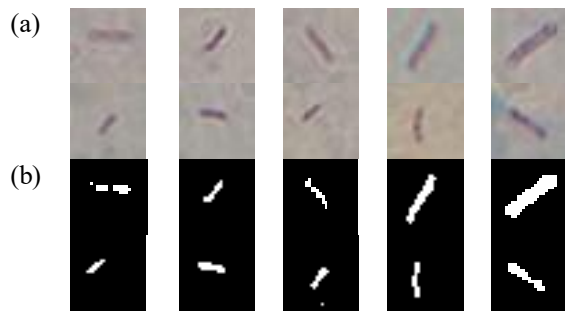
C.1 Dataset

Pada penelitian ini kami memilih menggunakan dataset dari NLM/NIH TB Dataset (USA) karena dilihat dari representasi citranya, citra bakterinya jelas (tidak blur, tidak bertumpuk dengan dahak, dan dahaknya tidak dominan warna biru), selain itu data MIDTI mempunyai kontras citra yang gelap (Gambar 6).



Gambar 6. Perbandingan Antara Citra (a) Dataset MIDTI dan (b) Dataset NLM/NIH TB

Dari dataset tersebut, diambil hanya 1 citra terbaik yang dipotong dengan dimensi 28x28. Citra yang dipotong hanya pada bagian yang ada bakterinya dan bagian kosong yang tidak ada bakterinya. Data patch yang diambil dari dataset NLM/NIH TB terdiri dari 10 citra patch bakteri dan 10 citra patch latar belakang, lengkap dengan citra labelnya yang diperlihatkan pada Gambar 7.



Gambar 7. Dataset Patch yang Digunakan (a) Citra bakteri, (b) Citra Latar Belakang, (c) Citra Label Bakteri, (d) Citra Label Latar Belakang

C.2 Pengujian

Pada pengujian dilakukan 2 hal untuk menjawab kontribusi, Dimana ada 2 kontribusi yang diusulkan yaitu tentang metode adaptive dan investigasi penggunaan operator logika.

1. Pengujian Model Adaptive

Pengujian model adaptive menggunakan perhitungan iterasi. Hal ini digunakan untuk melihat jumlah iterasi dan hasil kualitatif yang dihasilkan.

2. Investigasi penggunaan operator logika.

Pengujian dengan melakukan investigasi penggunaan operator logika, dilakukan dengan menggunakan confusion matrix (persamaan 11) dengan menghitung index IoU (persamaan 12) sebagai ukuran performansi system.

$$\begin{bmatrix} TP & FN \\ FP & TN \end{bmatrix} \quad (11)$$

$$IoU = \frac{TP}{TP+FP+FN} \quad (12)$$

Dimana TP adalah True Positif atau ketika objek terprediksi sama dengan GT. TN Adalah True Negatif atau ketika objek terprediksi berbeda dengan GT. Dimana FP adalah False Positif atau ketika non-objek terprediksi sama dengan GT. Dimana FN adalah False Negatif atau ketika non-objek terprediksi berbeda dengan GT.

IV. HASIL DAN DISKUSI

A. Hasil

A.1 Adaptive Iterasi

Pada iterasi adaptive, dilakukan pengukuran jumlah iterasi yang dibutuhkan untuk mencapai kondisi konvergen/optimal, Dimana kondisi optimal Adalah pada saat pixel objek (ditandai dengan biner "1") yang berdekatan saling terhubung. Hasil dari adaptive iterasi ditunjukkan pada Tabel 4.

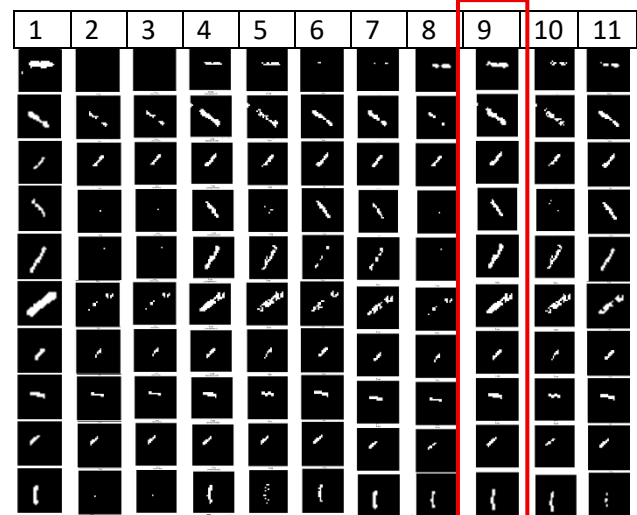
Tabel 4. Jumlah Iterasi Tiap Citra Tiap Operasi Logika

Operasi Logika	Citra Ke- (Jumlah Iterasi)									
	1	2	3	4	5	6	7	8	9	10
M-RGB	1	0	1	1	0	1	1	1	1	1
M-OR	1	1	0	1	1	1	1	1	1	1
M-AND	1	1	1	1	1	1	1	1	1	1
M-XOR	1	1	1	0	1	1	1	1	1	1
M-XAND	0	1	1	1	1	1	1	0	1	1

Pada Tabel 4 dapat dilihat bahwa operasi M-AND membutuhkan iterasi 1 kali pada keseluruhan citra, sedang operasi logika lain mempunyai jumlah iterasi bervariasi.

A.2 Investigasi Operator Logika

Pada pengujian investigasi operator logika, dilakukan proses pengamatan secara kualitatif dan kuantitatif.



Gambar 8. Hasil Annotasi Biner (1) GT, (2) RGB, (3) OR, (4) AND, (5) XOR, (6) XAND, (7) M-RGB, (8) M-OR, (9) M-AND, (10) M-XOR, (11) M-XAND

Pada proses pengujian kualitatif dilakukan dengan melihat langsung hasil annotasi biner yang dihasilkan (Gambar 8), sedangkan pada pengujian kualitatif dilakukan dengan menghitung nilai IoU rata-rata dari setiap operator logika yang digunakan. Pada Gambar 8, terdapat beberapa singkatan, Dimana "GT"=Ground Truth (Citra acuan), citra yang di binerkan dari citra warna dengan singkatan "RGB", citra hasil operator logika ditunjukkan dengan singkatan ("OR", "AND", "XOR" dan "XAND"), serta citra yang telah ditambahkan metode yang diusulkan menggunakan tambahan awalan M, yang artinya adalah modifikasi ("M-RGB", "M-OR", "M-AND", "M-XOR" dan "M-XAND"). Secara pengamatan pada Gambar 1, operator logika M-AND mampu memetakan keseluruhan citra (1-10) yang mirip dengan citra GT-nya.

Pada proses pengujian kualitatif dilakukan dengan menghitung nilai IoU rata-rata dari setiap operator logika yang digunakan yang ditunjukkan pada Tabel 4.

Tabel 4. Nilai Rata-Rata Tiap Operasi Logika

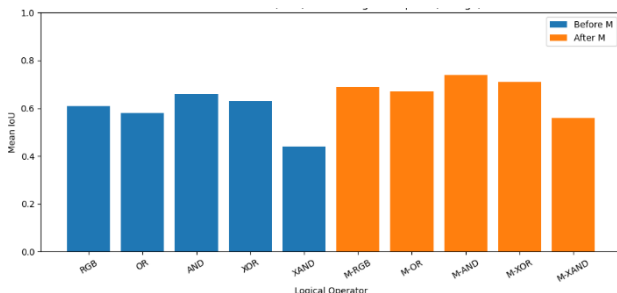
No	Operator Logika	Rerata IoU
1	RGB	0.61
2	OR	0.58
3	AND	0.66
4	XOR	0.63
5	XAND	0.44
6	M-RGB	0.69
7	M-OR	0.67
8	M-AND	0.74
9	M-XOR	0.71
10	M-XAND	0.56

Pada Tabel 4 dapat dilihat bahwa Ketika ditambahkan metode yang diusulkan, terdapat penambahan rerata IoU pada keseluruhan operator logika termodifikasi. rerata IoU tertinggi Adalah pada operator logika M-AND dengan rerata IoU mencapai 0.74.

B. Diskusi

Hasil eksperimen menunjukkan bahwa seluruh metode berbasis operator logika (RGB, OR, AND, XOR, XNOR) yang dilengkapi dengan mekanisme eigen-adaptive post-processing umumnya mencapai kondisi stabil dalam satu iterasi (Gambar 9). Namun, konsistensi iterasi ini hanya benar-benar terjaga pada metode M-AND di seluruh patch yang diuji, walaupun hanya terdapat peningkatan sebesar 23,33% performa antara sebelum modifikasi dengan setelah modifikasi. Metode lain menunjukkan variasi berupa anotasi kosong atau objek yang tidak utuh, fragmentasi berlebih, atau perubahan struktur graf yang tidak konsisten pada sebagian citra, meskipun secara numerik juga berhenti pada iterasi pertama.

Temuan ini mengindikasikan bahwa M-AND menghasilkan struktur graf awal yang sudah mendekati keadaan stabil, sehingga kriteria geometri berbasis nilai eigen Laplacian (khususnya λ_2) tidak memicu kebutuhan pemurnian lanjutan. Dengan kata lain, operator AND yang termodifikasi memberikan keseimbangan yang paling tepat antara reduksi noise dan preservasi konektivitas objek, sehingga graf mencapai fixed point lebih cepat dan konsisten.



Gambar 9. Grafik Performa Hasil Annotasi Sebelum Modifikasi (biru) dengan Sesudah Modifikasi (orange)

V. CONCLUSION

Karya ini menunjukkan bahwa penggabungan informasi geometri graf ke dalam alur segmentasi berbasis operator logika dapat secara substansial meningkatkan kualitas anotasi tingkat patch pada citra mikroskopis. Alih-alih memperkenalkan model berbasis pembelajaran yang kompleks, strategi adaptif eigen yang diusulkan memanfaatkan properti konektivitas graf intrinsik untuk memandu keputusan pasca-pemrosesan secara transparan dan efisien secara komputasi. Konsistensi konfigurasi M-AND yang diamati di seluruh perilaku iteratif, kemiripan visual dengan ground truth, dan kinerja IoU kuantitatif menyoroti pentingnya menyeimbangkan penekanan noise dan pelestarian struktur pada graf.

Selain analisis citra BTA, kerangka kerja yang diusulkan

dapat diterapkan secara luas pada tugas pencitraan biomedis lainnya yang melibatkan struktur latar depan yang jarang atau terfragmentasi, seperti analisis morfologi sel atau deteksi mikropartikel. Dimana citra medis sering kesulitan untuk mendapatkan anotasi dari dokter atau tenaga medis, sehingga kebutuhan akan anotasi otomatis un-supervise menjadi sangat menjanjikan dimasa depan. Pekerjaan di masa mendatang dapat mengeksplorasi integrasi konstruksi graf berbobot, agregasi patch multi-skala, atau penggabungan kriteria geometri yang diusulkan dengan model pembelajaran ringan untuk lebih meningkatkan performa hasil anotasi sambil mempertahankan kecepatan proses.

VI. APPENDIX

BTA	: Bakteri Tahan Asam
FP	: False Positif
FN	: False Negatif
GT	: Groud Truth (Citra Acuan)
XOR	: Exclusive OR
XAND	: Exclusive AND
RGB	: Red-Green-Blue (Citra Warna)
M- RGB	: Modified RGB
M- OR	: Modified OR
M- AND	: Modified AND
M- XOR	: Modified XOR
M- XAND	: Modified XAND
Non-Zero	: Jumlah selain angka 0 pada matrix
λ_2	: Eigen Value
TP	: True Positif
TN	: True Negatif

Explanation Video: <https://youtu.be/ghBApx0J49w>

VII. ACKNOWLEDGMENT

Ucapan terima kasih untuk dosen pengampu mata kuliah Aljabar Geometri atas bimbingan dan arahnya.

REFERENCES

- [1] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," International Journal of Computer Vision, vol. 59, no. 2, pp. 167–181, 2004, doi: 10.1023/B:VISI.0000022288.19776.77.
- [2] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," International Journal of Computer Vision, vol. 70, no. 2, pp. 109–131, 2006, doi: 10.1007/s11263-006-7934-5.
- [3] L. Grady, "Random walks for image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pp. 1768–1783, 2006, doi: 10.1109/TPAMI.2006.233.
- [4] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," International Journal of Computer Vision, vol. 104, no. 2, pp. 154–171, 2013, doi: 10.1007/s11263-013-0620-5.
- [5] J. A. Albano, D. W. Messinger, and S. R. Rotman, "Commute time distance transformation applied to spectral imagery and its utilization in material clustering," Optical Engineering, vol. 51, no. 7, 2012, doi: 10.1117/1.OE.51.7.076202.
- [6] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," IEEE

- Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 11, pp. 1101–1113, 1993, doi: 10.1109/34.244673.
- [7] E. Merkurjev, T. Kostić, and A. L. Bertozzi, “An MBO scheme on graphs for classification and image processing,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 4, pp. 1903–1930, 2013, doi: 10.1137/120886935.
- [8] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000, doi: 10.1109/34.868688.
- [9] S. V. Belim and S. Y. Belim, “Images segmentation based on cutting the graph into communities,” *Algorithms*, vol. 15, no. 9, 2022, doi: 10.3390/a15090312.
- [10] Z. Zhang, F. Xing, H. Wang, and L. Yang, “Revisiting graph construction for fast image segmentation,” *Pattern Recognition*, vol. 76, pp. 344–358, 2018, doi: 10.1016/j.patcog.2018.01.037.
- [11] A. Elmoataz, M. Toutain, and D. Tenbrinck, “On the p-Laplacian on graphs with applications in image processing and machine learning,” *SIAM Journal on Imaging Sciences*, vol. 8, no. 4, pp. 2412–2451, 2015, doi: 10.1137/15M1022793.
- [12] S. Li and D. O. Wu, “Modularity-based image segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 570–581, 2015, doi: 10.1109/TCSVT.2014.2360028.
- [13] N. Pourian, S. O. Arik, and T. Pfister, “Learning communities of image-parts for multi-class segmentation,” in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3278–3286, doi: 10.1109/ICCV.2015.160.
- [14] F. Verdoja and M. Grangetto, “Graph Laplacian for image anomaly detection,” *Machine Vision and Applications*, vol. 31, 2020, doi: 10.1007/s00138-020-01059-4.
- [15] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003, doi: 10.1162/089976603321780317.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025



Nama dan NIM